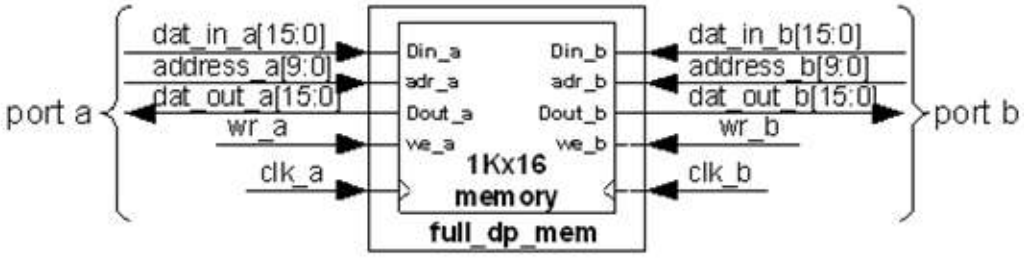
VHDL by Example, errata

We now introduce a few common state-type operations to show how increasingly sophisticated register-based functions are implemented in process statements. A four-bit counter is enabled by a "start" event, and stopped by a "stop" event. The SR flop allows the start and stop events to be short, e.g. one-clock pulses, rather than a continuously enabling flag. Additionally, for further illustration, we delay the ~~start~~ stop signal two clocks and provide it as an output.

You'll notice that we have not shown the asynchronous reset in the diagram. This is done for clarity; from this point forward it is assumed. It is implemented in the code, and always will be (in this book).

address simultaneously. There are two possibilities: 1) the data that is read is the original value before the write replaced it (called "read-before-write"), or 2) the data is the new value that is being written (called "write-before-read"). Our code implies a write-before-read operation based on the simple fact that the write assignment comes before the read assignment in the process statement (reversing the order would imply a read-before-write). Newer RAM blocks typically accommodate either type of operation, but some older versions are fixed, and in that case, as the designer you would have to make sure your code matches (and that your design operates correctly).

Next, we look at a full dual-port memory, where both ports have both write and read capability.



Full Dual-port Memory

It should be noted that vendors have different memory capabilities, and generally specific methods of inferring RAM. This simple example of write-before-read will not work for many vendor synthesis tools, and the designer is advised to consult the vendor documentation for guidance.